

Menus and Toolbars Section

Overview	Page 1
Menu Terminology	Page 1
Menu and Toolbar Standards	Page 2
Menu and Toolbar Guidelines	Page 9
Recommended Readings	Page 12
Helpful Web Sites	Page 13

Menus and Toolbars

Overview

The modern GUI would not be complete without menus and toolbars. Menus are the best way of showing beginning users how to use the application. Toolbars provide the advanced user with the quickest and easiest interaction - just move the mouse over the toolbar button and click. While toolbars can also benefit the beginning user, they are not quite as effective as the menu. Menus provide context to the beginning user. Menus are the showcase to all of the features that are available in the application. When the user clicks on the menu, a drop down menu appears. All of the commands in the drop down menu are associated with the menu. This gives the user clues as to how the commands are organized and what area of functionality they belong to. Toolbar buttons provide the user with none of this information and, as a result, are less comfortable for the beginning user.

This document is primarily aimed at Windows software development. While menus are also implemented on the web, they are implemented differently than is described in this document. Web developers should follow the general guidelines that are outlined here for menus, keeping in mind that they may implement them differently. In addition, web menus do not typically have the same shortcut and access key requirements that are used in Windows.

Menu Terminology

A *menu bar* is the area displayed across the top of the window directly below the window title bar. The menu bar is made up of entries called *menu titles*. When the menu title is clicked on, a *dropdown menu* is shown. Each entry in the dropdown menu is called a *menu item*. Menu items can also have sub-menus that provide the user with access to additional choices. These additional choices are known as a *cascading menu* and are recognizable from the drop down menu because of the triangular arrow that appears next to the menu item that they belong to.

Another category of menu is the *pop-up menu*, which provides an efficient way for the user to access the commands that are associated with a specific object. They are initiated by pointing to the object and using the alternative (button-2) button of the mouse. Since the pop-up menu is displayed at the pointer's current location, they eliminate the need for

the user to move the pointer to the menu bar or toolbar. They also put the command in immediate context to the object, reducing the number of commands the user must browse through. Once again, realize that pop-up menus are more useful for the advanced user.

Menu and Toolbar Standards

The following menu and toolbar standards are to be adopted across all CDC Windows software applications and web-based applications. These standards should lead to a more consistent and usable interface.

Menus

Use single word menu titles without hyphens, slashes, or other word delimiters

Multiple word menu titles look like multiple menus. Also, do not use hyphens, slashes or other word delimiters since they make the menu bar look awkward.

Use conventional book title capitalization in all menu items

Capitalize the first letter of each word in menu items unless it is an article or preposition not occurring at the beginning or end of the name, or unless the word's conventional usage requires it to not be capitalized.

Left justify all menu titles and menu items

Right justification of menus is possible, but is considered obsolete, and makes the menu hard to read.

Use a single menu bar

Menu bars that are so long that they have to take up multiple lines in ordinary use are an indication that the menu is too complex. If the menu appears too complex, consider moving some of the commands to dialog boxes.

Do not use multiple column drop down menus

Multiple columns add unnecessary complexity to menus.

Disable invalid menu items

Since menus give the user a summary of what functions are available, it's important to keep menus and menu items stable. Do not remove invalid menu items; rather, disable them. If the menu item will never be available during a session of the program, then remove it.

Do not give errors on invalid menu commands

Menus should not contain any commands that will only result in an error message. Disable those commands instead.

Do not use "Bang" menu titles

Bang menus are menu selections in the menu bar that act as a command, instead of dropping down a menu. These should never be used.

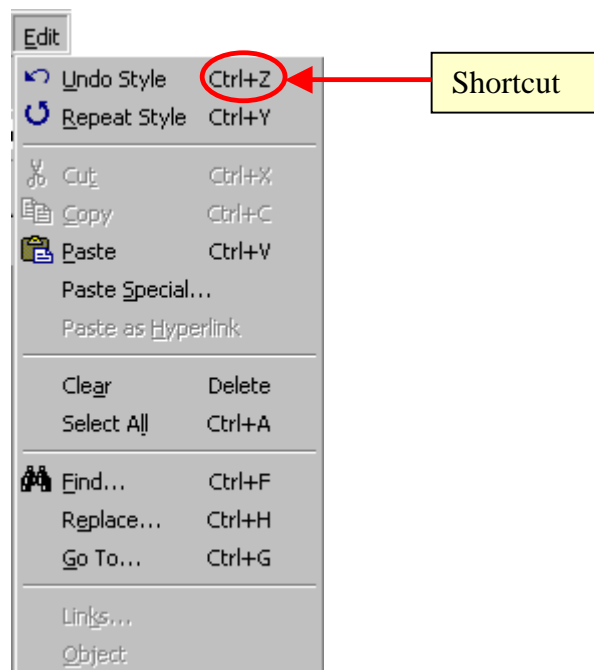
Provide access keys on all main menu titles and menu items

Access keys provide keyboard access to components of the menus and should be assigned to all menu titles and menu items. Access keys must be unique across the menu bar and within a dropdown menu. For more information on access keys, see the Navigation section of this style guide.

Display all shortcuts in the menu item

Shortcuts, otherwise known as hot keys, or accelerators, allow the user to perform commands directly from the keyboard, and are very efficient for the user if they already know the shortcut command. To facilitate learning, all keyboard shortcuts for menu commands will be displayed to the right of the menu item as shown in the figure titled *Menu Shortcuts*. For more information on how to establish shortcut keys, see the Navigation section of this style guide.

Menu Shortcuts

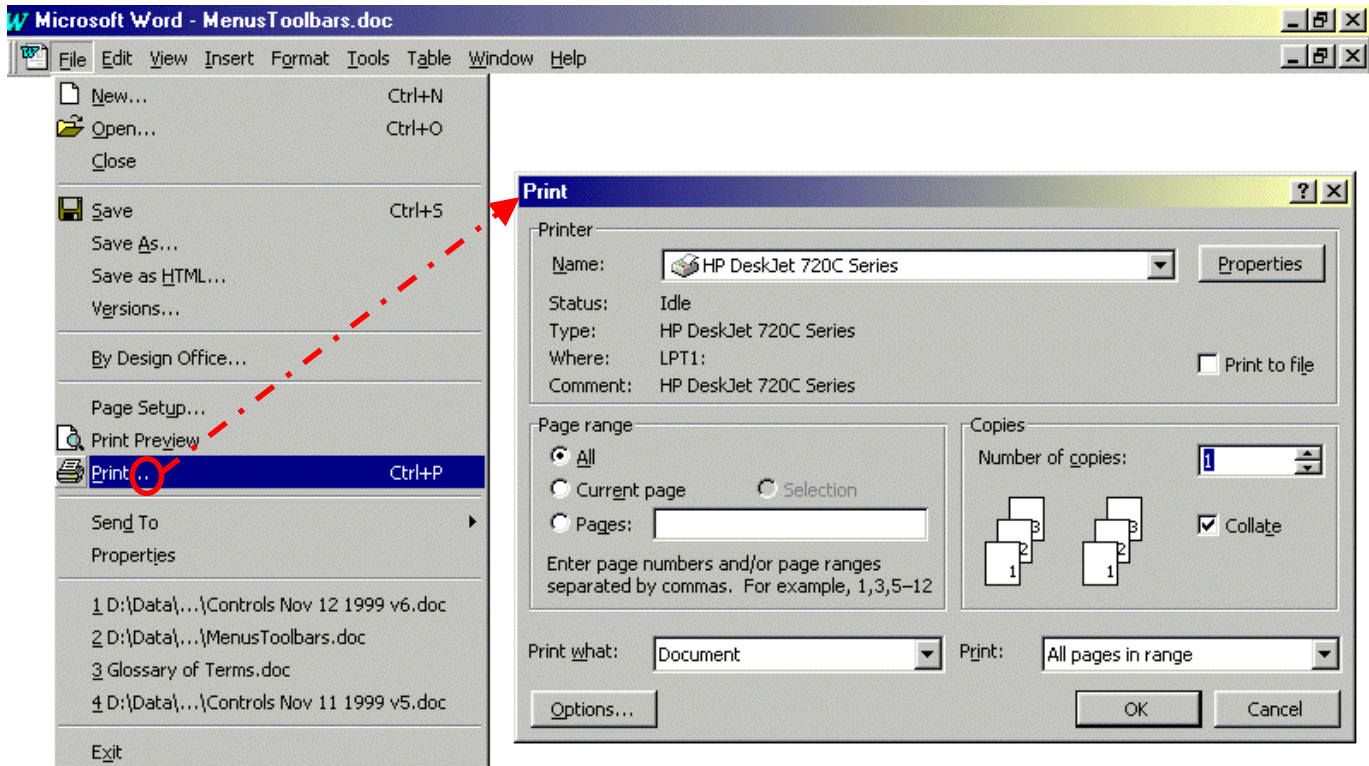


Use ellipses to indicate that more information is required

Use ellipses to indicate that more information is required in order to carry out the command. The figure titled *Menu Item Ellipses* illustrates the use of the ellipse

on the Print command to indicate that additional information must be gathered from the user in order for the command to be carried out.

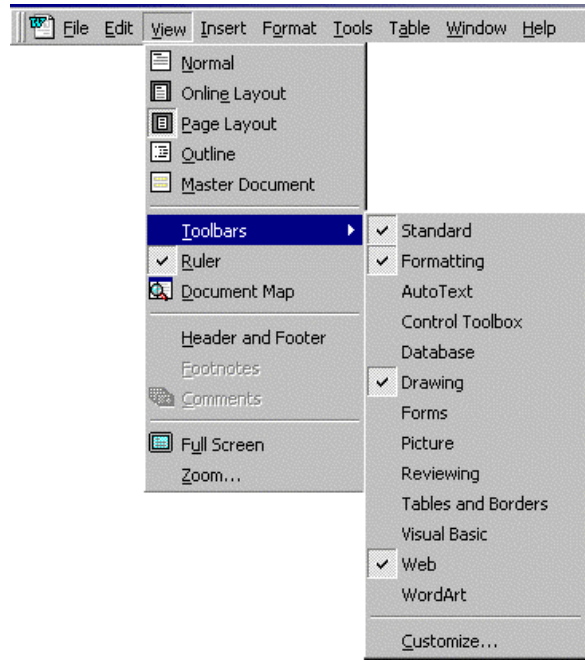
Menu Item Ellipses



Use check marks on menu items to turn options on and off

Check marks are very effective when turning options on and off and should not be used to change to any other state for menu items. The figure titled *Menu Item Check Mark Example* illustrates the use of check marks to turn the Ruler command on, as well as, to turn on various toolbars in the Toolbar cascading menu.

Menu Item Check Mark Example



S All commands available in a pop-up menu will also be available on the menu bar

Pop-up menus, otherwise known as context menus, will not be the only way to access a command. Pop-up menus make command access more efficient for advanced users.

S Do not use access keys in pop-up menus

As stated earlier, pop-up menus are for quick access and are accessed with the mouse only, not the keyboard.

S Use right pointing triangles for cascading menus

A cascading menu will be shown to the user with a black triangle that points to the right as shown in the figure titled *Cascading Menu Prompt*.

Cascading Menu Prompt





Use standard menus

Users expect File, Edit, Windows, and Help menus in Windows applications. For this reason, you should supply those menus in all Windows applications. In addition, place standard menu items in these menus. For example, users expect to see Cut, Copy, and Paste in the Edit menu. Similarly they expect to see Print and Exit under the File menu even though they have nothing to do with Files, and they should see at least the About command under Help. See the table titled *Windows Standard Menus* for a description of the standard menus.

Windows Standard Menus

Menu Title	Description
File	Provides primary operations that apply to a file or data type. Make sure to include commands such as Open, Save, Print, and Exit. Exit should always be placed at the very bottom of the menu.
Edit	General purpose editing commands such as Cut, Copy, and Paste. The following commands should also be included if they are supported by the application: Undo, Repeat, Find and Replace, Delete, Duplicate.
Window	Used in a multiple document interface style only. Includes commands associated with managing the child windows. Always include a window list in the Window menu.
Help	Provides commands that provide access to help information. Include a Help Topics command for access to the standard Help browser. You can also provide commands to access specific pages of the Help Topics browser, such as Contents, Index, and Find Topic. Place other user assistance commands in this menu that are specific to your application.

There are other Windows optional menus that, when included in your application should contain specific commands in order to be consistent with other Windows application. See the table titled *Windows Optional Menus* for a description of other common menus.

Windows Optional Menus

Menu Title	Description
View	Provides commands that change the user's view of data within the window, but not the data itself. Also include commands for controlling the display of particular interface elements - like Toolbars, Show Ruler.
Insert	An extension of the Edit menu. If you only have a couple of commands in this menu, then consider placing them in the Edit menu.
Tools	Sometimes called options or functions, these are powerful transformation functions. These functions are typically used by power users.

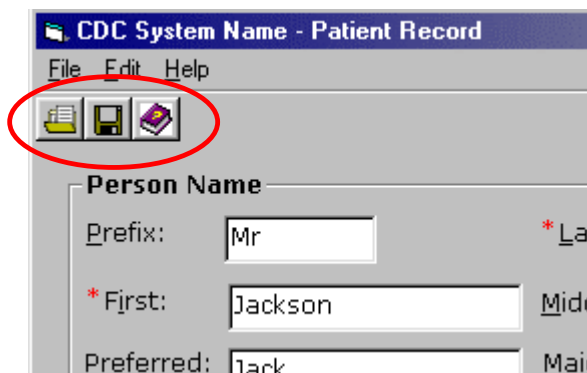
Toolbars



Use small toolbar buttons

Because the number of features in an application can get large, it is better to use small toolbar buttons as illustrated in the figure titled *Toolbar Example*. Graphic images for small toolbar buttons should be no larger than 16 by 16 pixels.

Toolbar Example



Do not include captions on toolbar buttons

The toolbar image and the tooltip should be enough to convey the command to the user. Do not include static text within the toolbar button.

Disable toolbar buttons or remove entire toolbar

Invalid toolbar buttons should be disabled rather than removed. However, consider removing the entire toolbar when the user enters a mode where nothing on the toolbar will be available.

Do not give errors on invalid toolbar buttons

Toolbars should not contain any commands that will only result in an error message. Disable those toolbar buttons where the associated command is invalid.

Always provide tooltips

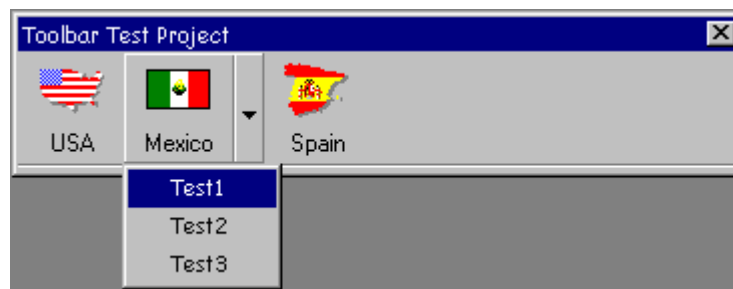
Tooltips help the user understand the meaning of the toolbar button. Include tooltips on all toolbar buttons. When creating tooltips, keep in mind the following:

- ***Make tooltips informative but brief***
- ***Aim tooltips at intermediate to advanced users***
- ***Use tooltips to remind the user, not to teach***
- ***Include shortcut key designations when there is a shortcut key***

Do not use dropdown toolbar buttons

Although many development tools now allow the creation of dropdown toolbar buttons as shown in the figure titled *Drop Down Toolbars*, CDC software applications should make all toolbar button action immediate. Thus toolbar buttons should initiate a command directly and not provide drop down menus.

Drop Down Toolbars
(avoid these)



Do not display tooltips in the status bar

For consistency purposes, avoid displaying tooltip text information in the status bar at the bottom of the screen.

***Use the same tool bar icons***

Common tool bar icons will be used across all CDC-developed applications. See the Graphics section of the CDC User Interface Style Guide for these icons.

Menu and Toolbar Guidelines

The following guidelines have been established within the Windows user interface design community and should be applied across CDC software applications and web-based applications.

Menus***Balance dialog boxes and menus***

Menus become complex when there are too many menus in the menu bar or when there are too many menu items in the menu drop down. If the menu structure becomes too complex, consider moving some of the commands into a dialog box. On the other hand, if your menu structure appears too simple, remove some redundant dialog boxes and place those commands into the menu structure.

Order menu items appropriately

More important and frequently used commands should appear at the top of the menu, less important and seldom-used commands should appear at the bottom of the menu.

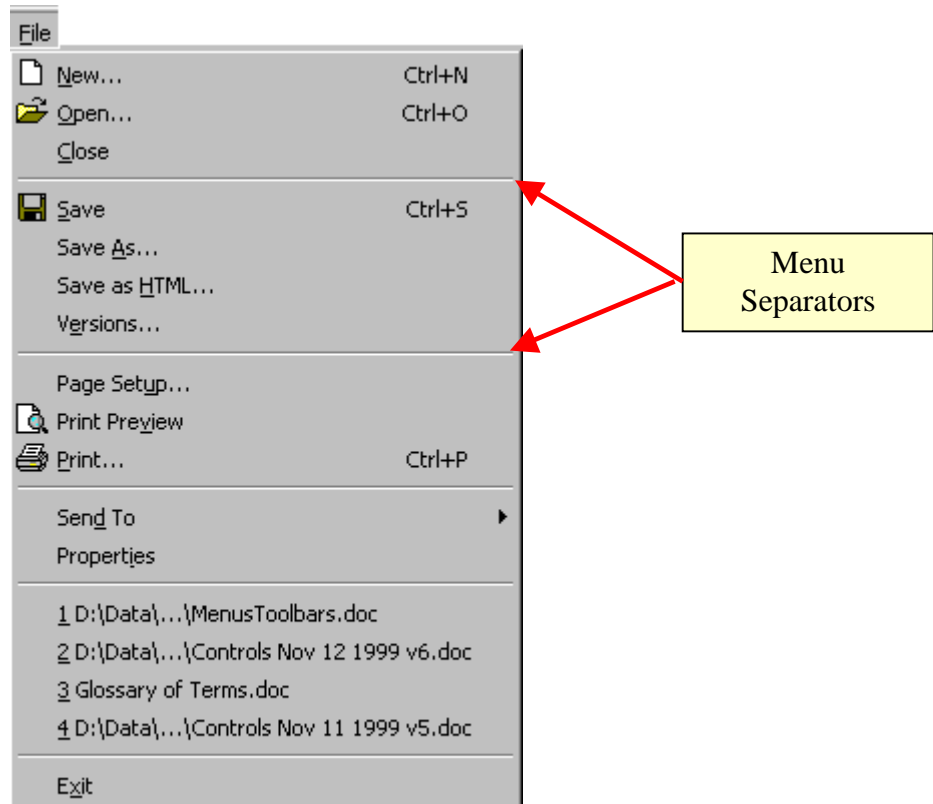
Organize clearly related menu items into groups

Grouping menu items can greatly simplify menus. Menu separators and cascading menus are the most common techniques for grouping menu items.

Include 2 or more menu items between menu separators

To avoid having too many menu separators, avoid having groups with only one item in them. This may involve creating a group of unrelated items as Microsoft did in the File menu when they grouped Send To and Properties in the same group even though they were unrelated. The only exception to having a group contain a single item is when the menu item is the last item in the drop down and it is necessary to make it stand out from the rest of the menu items in the drop down. An example of this is the Exit menu item in the File menu. See the figure titled *Using Menu Separators* for an illustration of menu separators and groups.

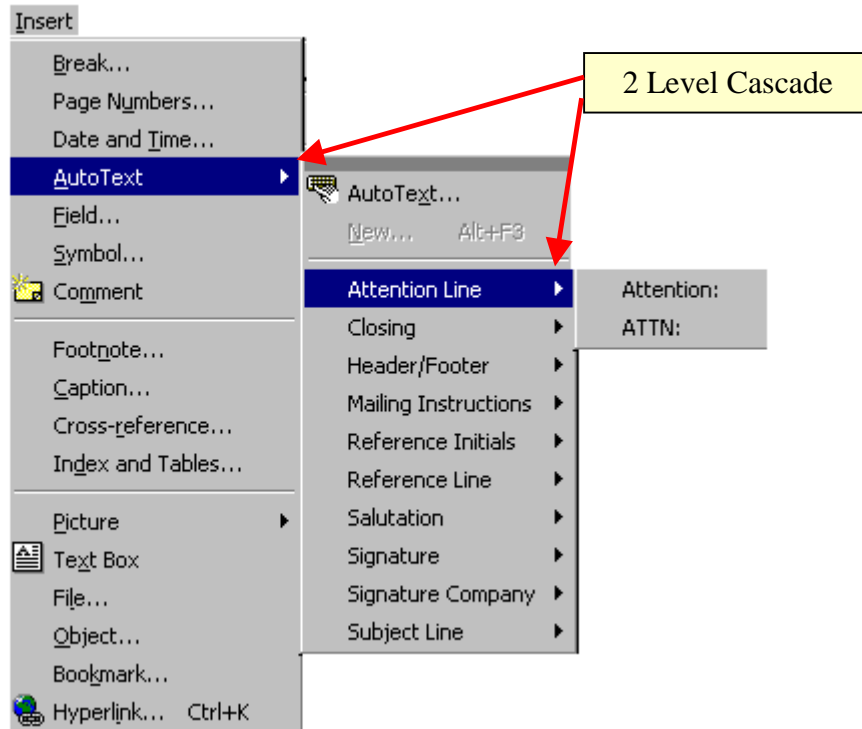
Using Menu Separators



Avoid cascading menus for frequently used commands and limit their levels

Cascading menus are a technique used for grouping menu items as shown in the figure titled *Cascading Menu Groups*. They are best used to allow the selection of one of a set of related options. Since cascading menus are difficult to use, they should be limited to one or, at most, two levels of submenus.

Cascading Menu Groups



Toolbars

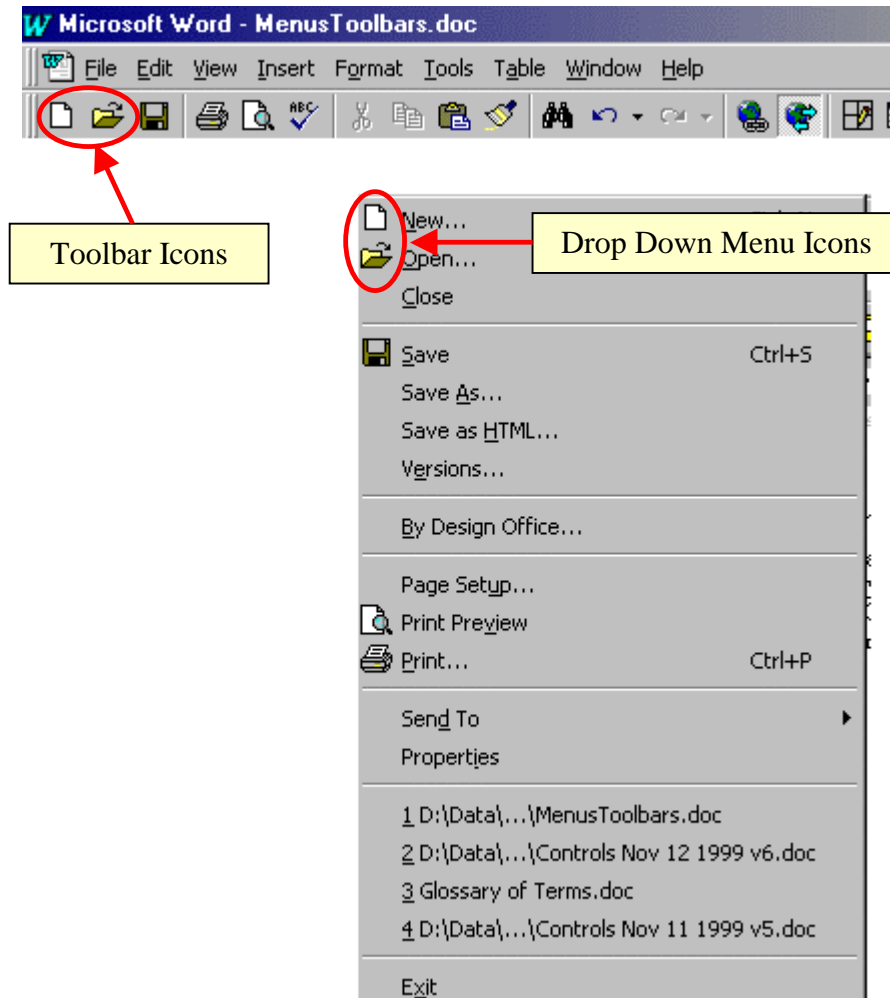
Avoid user-configurable toolbars whenever possible

Toolbars should be docked at a screen location below the menu bar, at the top of the screen. The toolbar should not be configurable by the user – no customized items are allowed on the toolbar.

Provide reference between toolbar and menu when possible

As was indicated earlier, toolbars are targeted more for the advanced user. However, they can provide increased productivity for all users once the user learns the toolbar. One way to assist the user in learning the toolbar is to include toolbar icons in the menu items as shown in the figure titled *Matching Toolbar Icons and Menu*. Some development tools may require third party controls to accomplish this.

Matching Toolbar Icons and Menu



Recommended Readings

McKay, Everett N. *Developing User Interfaces for Microsoft Windows*. Microsoft Press, 1999

Chapter 4 - Establishing a Consistent User Interface Style

Chapter 6 - Beginning vs. Advanced Users

Chapter 10 - Good Interfaces are Invisible

Chapter 15 - Keep It Simple

Chapter 21 - Tooltips are Cool

Chapter 22 - Unnecessary Dialog Boxes are Evil

The Windows Interface Guidelines for Software Design. Microsoft Press, 1995

Chapter 7 - Menus, Controls, and Toolbars

Helpful Web Sites

The Microsoft Developers Network Online Library of Books, specifically the online version of *The Windows Interface Guidelines for Software Design*
<http://msdn.microsoft.com/isapi/msdnlib.idc?theURL=/library/books/winguide/PLATFRM2/D5/S115B5.HTM>

MacIntosh Human Interface Guidelines Online version Chapter 4 - Menus
<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-75.html#HEADING75-0>

Human Factors International
online article titled "*Pull-Down Menus: Out of Sight, Out of Mind*"
<http://www.humanfactors.com/library/menus.asp>